

Ejercicio 1:

1) a) Cada letra lo mencionamos a un gráfico de tiempos: a- escritura, b- lectura, c- lectura, d- escritura,

e- lectura, f- escritura (nos basamos en el valor de la señal r/\bar{w} durante la operación)

b) La transferencia involucra toda la operación de lectura y escritura (tanto de la dirección de memoria con la señal req y ack / tanto de los datos, atendiendo a la señal ack).

Por diagrama de tiempo tenemos: a-4, b-4, c-4, d-4, e-4, f-4

c) En todos los gráficos, las líneas clk , req , r/\bar{w} y ack son dedicadas (se usan con un solo propósito y tienen un comportamiento de igual manera). En los gráficos a- y b- la línea data es multiplexada (transfiere datos y direcciones de memoria), mientras que en el resto no lo es por la presencia de la línea dir (dedicada a direcciones).

Para los gráficos e- y f- la línea mem/der es también dedicada (indica si se establece la operación con la memoria o uno de los dispositivos).

d) a- data = 12 (pues el ancho de la palabra es de 12b, más alto de que la línea puede usarse para transferir direcciones de memoria) b- data = 12 (idem a)

e- data = 12, dir = 7 (a la línea de datos le asigno el ancho de palabra, mientras que a la de direcciones le asigno la cantidad de bits que permite direccionar)

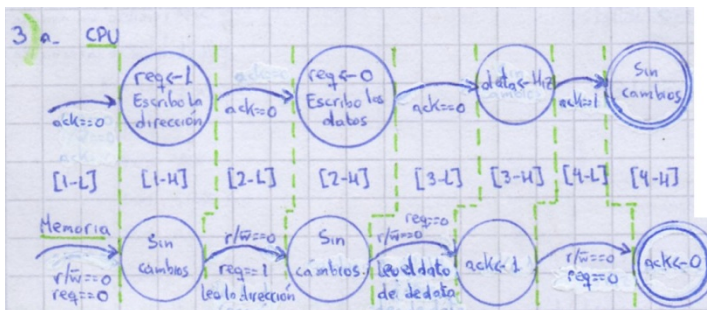
d- data = 12; dir = 7 (idem c) e- data = 12; dir = 7 (idem c) f- data = 12; dir = 7 (idem c)

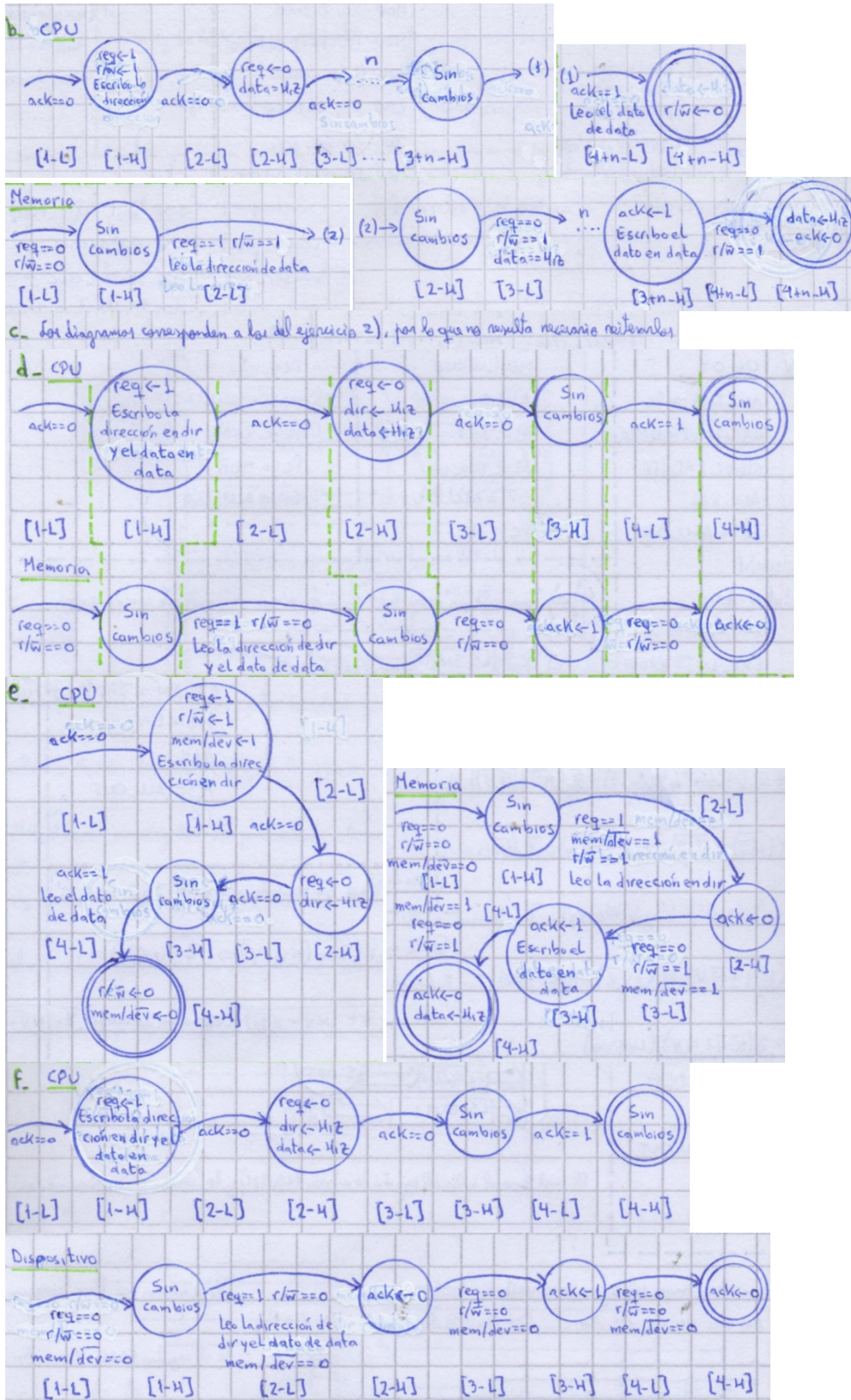
e) En los gráficos a-, b-, c-, d- parece haber un único espacio de direcciones pues las comunicaciones se establecen entre la CPU y la memoria, y en las líneas no se hace una distinción según la dirección enviada. Sin embargo, en los gráficos e- y f- aparece la línea mem/der con el propósito de distinguir si la operación se realiza hacia un dispositivo o a la memoria, y como se transfiere una dirección por dir, vemos que los dispositivos poseen un espacio de memoria reservado.

Ejercicio 2:

2) Dichos rangos de estado se corresponden con el diagrama c- del ejercicio 1. Esto lo podemos ver viendo que la CPU envía la dirección en el estado 1 y la pone alta impulsando en el cuarto, pero no en el quinto, con lo cual se tiene una línea dedicada a direcciones que transfiere una dirección durante 3 ciclos (lo cual nos sirve a todos los diagramas salvo el e-).

Ejercicio 3:

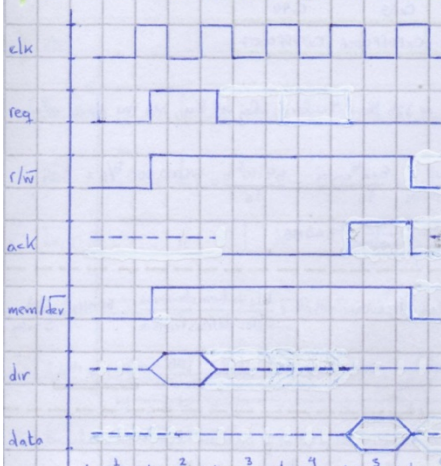




Ejercicio 4:

4) Sabiendo que el clock posee una frecuencia de 50 KHz o 50.000 $\frac{\text{ciclos}}{\text{s}}$ no para los diagramas a-, c-, d-, e- y f- que $50.000 \frac{\text{ciclos}}{\text{s}} \rightarrow 1 \text{ s} \rightarrow t = \frac{4}{50.000} \text{ s} = 80 \text{ ns}$.
 Pero el diagrama b- requiere saber lo que tarda un ciclo: $4 \text{ ciclos} \rightarrow 80 \text{ ns}$ Con esto, tengo que tarda $80 \text{ ns} + n \cdot 20 \text{ ns}$
 $1 \text{ ciclo} \rightarrow t = \frac{80}{4} \text{ ns} = 20 \text{ ns}$

Ejercicio 5:

5a) El bus estará conformado por los siguientes líneas dedicados: la control tenemos req (para que la CPU haga una petición), clk (para el bus en sí mismo), ack (para que la CPU sepa que la petición fue recibida y procesada), r/w (para que la CPU indique si la operación es de lectura o escritura), mem/dec (para que la CPU distinga si se desea comunicarse con la memoria, o uno de los dispositivos de E/S). Para los datos tendremos la línea dedicada data de ancho 32b (una palabra) y otra dir para direcciones de memoria.
 Sabiendo que el espacio direccionable es 1GB para direcciones de memoria, y la unidad direccionable es de 32b, luego se necesitan $\log_2 \left(\frac{1GB}{32b} \cdot \frac{8 \cdot 2^{30} b}{1GB} \right) = \log_2 \left(\frac{2^{33}}{2^5} \right) = \log_2 (2^{28}) = 28 \text{ bits}$ para distinguir los cables de memoria; sabiendo que el espacio de direcciones para los dispositivos es de 1MB = $2^3 \cdot 2^{20} b = 2^{23} b$, como no caben en el espacio que ocupan mis registros, tomaremos que deben ser de mínimo 1b, por lo que se necesitan al menos $\log_2 \left(\frac{2^{23} b}{1b} \right) = 23 \text{ bits}$ para distinguirlas. Usaremos 28b para ellos (que damos direcciones de dispositivos sin usar) para que sean consistentes con los de memoria, quedando la línea dedicada dir de ancho de 28b.
 b) Sabiendo en cuenta que los ciclos de reloj del bus duran 35ns y la memoria tiene un retardo de 70ns desde que lee del bus que la hacemos una petición, tenemos:

 Nota que entre que la memoria lee que la hacemos una petición hasta que carga el dato en la línea data y le envía un 1 en ack para confirmarnos que la petición fue procesada, hay un retardo de 70ns, lo cual equivale a 2 ciclos de reloj. A su vez, podemos tomar por protocolo que los líneas se juntan en el ciclo bajo del clock, y se cargan en el alto. Algo que debe considerarse es que la línea ack de la está en alta impedancia hasta que la memoria lee que la CPU quiere hacer una petición. Esto es porque hay más de un dispositivo que puede tomar el rol de slave y se evitan estos conflictos de solicitudes.

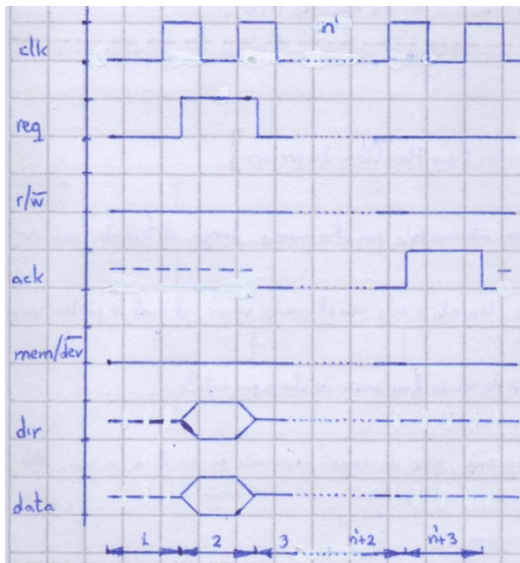
dos colisiones, hacen llegando a cuando se conectan dos dispositivos que simultáneamente envían señales de datos, siendo que ese nivel es 0 o 1.

l) Tomado eléctricamente como el potencial eléctrico o diferencia de tensión con respecto a un punto tierra (0V), estos pueden diferir en dicho punto tomado como referencia, y de ahí la entrada 0 de una señal puede interpretarse como un 1 en un dispositivo y viceversa, produciendo un resultado inesperado.

Esto se arregla desconectando aquel dispositivo no tomados en cuenta en una instancia a través de compuertas de 3 estados o multiplexores.

c) Ahora, para realizar la escritura en un dispositivo, tomamos que estos tienen un tiempo X. Sea

q, tomaremos el cociente de dividir X por 35 ns, para saber a cuántos ciclos equivale, y si el resto no es 0, le sumaremos 1 (no lo pudo cargar en los mini-ciclos dñs):



Tomamos entonces que n' es la cantidad de ciclos que necesita el dispositivo para localizar la celda en base a la dirección de memoria recibida (en el aspecto de direcciones de los dispositivos) y escribir el dato recibido en la misma, una req que haya que recibir una petición. Nótese que usamos n' para distinguirlo de n (cantidad de dispositivos de E/S) y que la señal ack comienza y termina en alta impédancia (desconectada).

Ejercicio 6:

6) a) Siendo que el bus es asimétrico, no se puede tener una señal de clk para sincronizar los señales. Por ello debemos describir una sucesión de eventos concatenados uno tras de otro para realizar una operación, los cuales se basarán en las señales de control. Estas serán las siguientes:

req: señal de petición que suena al master para indicarle al slave que desea realizar una petición

r/w: señal que denota el tipo de operación a realizar (read=1, write=0) ack: señal que avisa al slave para indicar que la petición fue recibida, junto con el tipo de operación. Nótese que no hace falta tener una señal de control que distinga si la operación se realiza con la memoria o uno de los dispositivos de E/S pues tomaremos que estos poseen registros mapeados a memoria. Además tendremos las líneas dedicadas data y dir para representar las líneas de datos y de direcciones respectivamente.

b) El ciclo de bus para escribir el valor 0x4453366 a partir de la posición 0xFFFF0004 debería ser como el que sigue:

1. Master carga 0xFFFF0004 en dir, 0x4453366 en data y baja la señal \overline{CS} .

2. Master levanta la señal \overline{req} .

3. Slave se que le está haciendo una petición y luego de un tiempo determinado en el que escribe los datos en memoria, levanta la señal \overline{ack} .

4. Master se que la señal \overline{ack} fue levantada y pone dir y data en alta impedancia (deja de enviar la dirección y los datos) y baja la señal \overline{req} .

5. Slave se que la señal de \overline{req} bajó y baja la señal \overline{ack} , terminando la operación.

Siendo que los datos se almacenan en modo little endian, el dato posee 32b y el direccionamiento es a byte (8b), tenemos que a partir de la carga de dirección 0xFFFF0004, el contenido de los multiplexantes vale:

Contenido	0x66	0x33	0x55	0x44
Dirección	0xFFFF0004	0xFFFF0005	0xFFFF0006	0xFFFF0007

Ejercicio 7:

7) Siendo el procesador de 32b, el bus de datos de 32b, el clock del bus de 8MHz y la transferencia de 32b lleva 4 ciclos de reloj del bus, ¿es que puedo calcular

$$\text{la máxima capacidad del bus como: } \text{máx capacidad} = \frac{\text{cont. bits transferidos}}{\text{ciclos de transferencia}} \times \text{frecuencia del clk} = \frac{32b}{4 \text{ ciclos}} \times \frac{8 \times 10^6 \text{ ciclos}}{1s} = \frac{64 \times 10^6 b}{1s} = 64.000.000 \text{ b/s} =$$

$$= \frac{64.000.000 b}{1s} \times \frac{1B}{8b} \times \frac{1KB}{1024B} \times \frac{1MB}{1024KB} \approx 6.25 \text{ MB/s}$$

Ejercicio 8:

8) Siendo el bus sincronizado de 50MHz, 32 líneas de direcciones y datos:

a) Se tiene que 50.000.000 ciclos — 1s

$$1 \text{ ciclo} \longrightarrow t = \frac{1}{50.000.000} s = 0,02 \times 10^{-6} s = 20 \text{ ns}$$

Ampl. máx capacidad: $\text{frecuencia del clk} \times \frac{\text{bits de transferencia}}{\text{ciclos de transferencia}} = 50 \text{ MHz} \times \frac{32b}{5 \text{ ciclos}} =$

$$= \frac{50.000.000 \text{ ciclos}}{1s} \times \frac{32b}{5 \text{ ciclos}} = 320 \times 10^6 \frac{b}{s} \times \frac{1B}{8b} \times \frac{1KB}{1024B} \times \frac{1MB}{1024KB} \approx 31,25 \text{ MB/s}$$

b) El bus debería al menos tener los líneas que hacen que al master le indique al

slave que se desea realizar una transferencia (llamase \overline{req}) y otra que al slave le indica al master que la operación fue procesada (llamase \overline{ack}). Según si la operación de transferencia es siempre la misma o no, puede que una señal que indique el tipo de operación no sea necesario ($\overline{r/w}$).

Ejercicio 9:

9) a) Este bus debería tener 16 líneas tanto de direcciones como de datos, debido al ancho de las direcciones y de la palabra. A su vez, el bus debería tener las líneas de control \overline{MREQ} (para indicar que se realiza un acceso a memoria), \overline{CLK} (para el bus es sincronizado), \overline{RD} (para indicar si la operación es de lectura o escritura), \overline{WAIT} (para que el slave se comuniquen con el master, indicándole que está procesando la operación, o ya terminó). No hacen falta nuevas líneas de control.

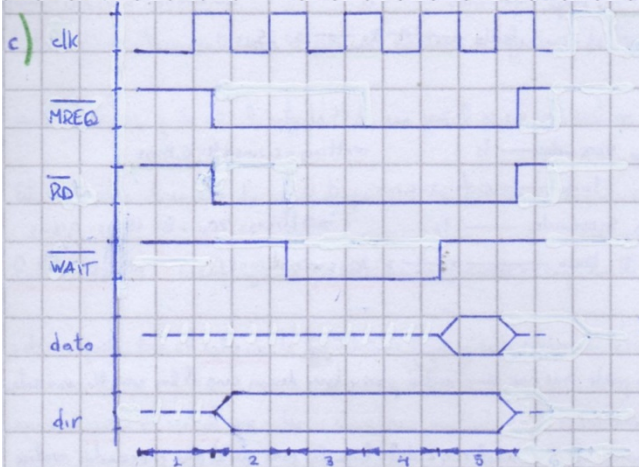
b) Tomamos por protocolo que al iniciar la lectura los bus de datos y direcciones se encuentran en alto impedancia, y los señales \overline{RD} , \overline{MREQ} y \overline{WAIT} se encuentran levantados. Continuamos el protocolo anterior con el paso 9:

9- Una vez que la CPU lee los datos, pone el bus de direcciones en alto impedancia pues ya no necesita indicar la dirección.

10- La CPU suelta la línea \overline{RD} pues ya lee los datos.

11- La CPU suelta la línea \overline{MREQ} indicando que los datos fueron leídos y se debe terminar la transferencia.

12- El módulo de memoria pone el bus de datos en alto impedancia una vez que la CPU lee los datos (lo cual tomaremos que puede hacerse en un ciclo).



Una transferencia entre la memoria y la CPU se realiza en 5 ciclos de reloj.

d) Siendo que por transferencia se transfieren 16b, cada transferencia lleva 5 c

iclos de reloj y cada ciclo es de t_{ns} , res que $8B = 64b = 4 \times 16b \Rightarrow$ deben reali

zarse 4 transferencias y res que:

1 transferencia — 5 ciclos

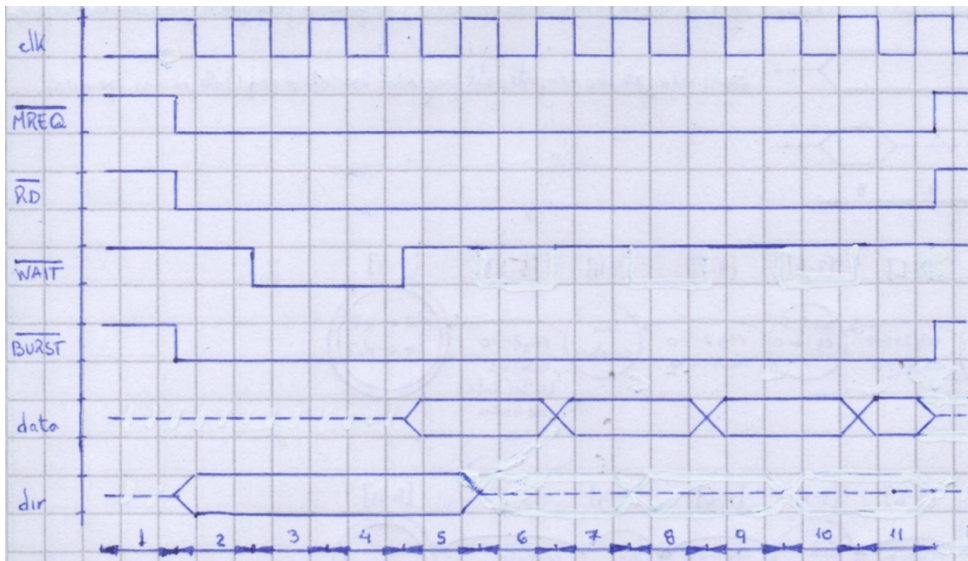
4 transferencias — $x = 5 \times 4 = 20$ ciclos

Y a su vez:

1 ciclo — t_{ns}

20 ciclos — $x = 20 \times t_{ns}$

e) Con el agregado de la línea \overline{BURST} cambiamos el protocolo inicial que en el caso de que \overline{BURST} malga 0 (nifago), podemos decir que los señales \overline{MREQ} y \overline{RD} se levantan una vez que se lee la cuenta palabra, y siendo que los palabras se cargan cada 2 ciclos, la señal \overline{WAIT} puede levantarse luego de lo primero.



Observa, siendo que la transferencia toma 11 ciclos en ejecución, y por la consigna cada ciclo es de 1ns, se infiere que una transferencia toma 11 ns. Véase que los 4 palabras son consecutivos a la primera, por lo que basta con que la dirección de memoria se indique para la primera.

Ejercicio 10:

10) Para calcular la capacidad de cada bus, nota primero que la lectura toma 8 ciclos en el protocolo serie, y 4 en el paralelo. Con esto, sea que:

a) PAR: frecuencia = 300 Hz \Rightarrow máx capacidad = $\frac{\text{bits transferidos}}{\text{ciclos de transferencia}} \times \text{frecuencia} = \frac{3b}{4 \text{ ciclos}} \times \frac{300 \text{ ciclos}}{1s} = 225 \text{ b/s}$ (00) \Rightarrow 3b transferidos. Con su vez, vemos que $\frac{300 \text{ ciclos}}{1s} \Rightarrow$ y luego tiempo de transferencia = $3,3 \text{ ms} \times 4 \text{ (ciclos)} = 13,3 \text{ ms}$

SERIE: frecuencia = 600 Hz \Rightarrow máx capacidad = $\frac{\text{bits transferidos}}{\text{ciclos de transferencia}} \times \text{frecuencia} = \frac{3b}{8 \text{ ciclos}} \times \frac{600 \text{ ciclos}}{1s} = 225 \text{ b/s}$. Luego $\frac{600 \text{ ciclos}}{1s} \Rightarrow$ tiempo de transferencia = $1,6 \text{ ms} \times 8 = 13,3 \text{ ms} \Rightarrow$ ambos protocolos también lo mismo

b) PAR: f (frecuencia) = 10 KHz \Rightarrow máx cap = $\frac{\text{bits trans.}}{\text{ciclos trans.}} \times f = \frac{3b}{4 \text{ ciclos}} \times \frac{10 \times 10^3 \text{ ciclos}}{1s} = 7500 \text{ b/s} = 937,5 \text{ B/s}$. Luego, $\frac{10.000 \text{ ciclos}}{1s} \Rightarrow$ tiempo de transferencia = $100 \mu s \times 4 = 400 \mu s = 0,4 \text{ ms}$

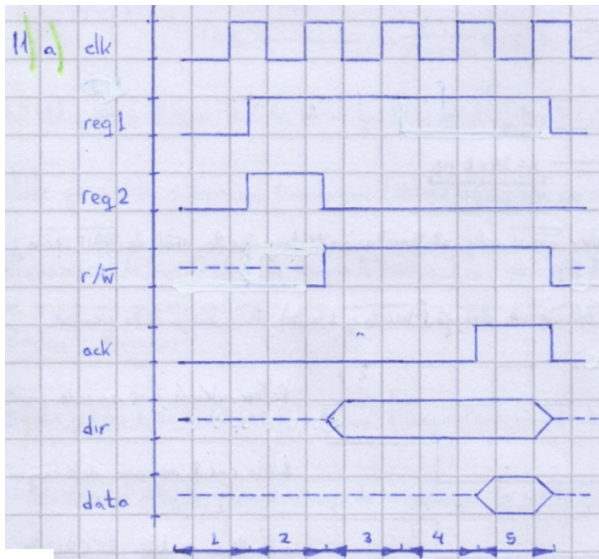
SERIE: f = 40 KHz \Rightarrow máx cap = $\frac{\text{bits trans.}}{\text{ciclos trans.}} \times f = \frac{3b}{8 \text{ ciclos}} \times \frac{40 \times 10^3 \text{ ciclos}}{1s} = 15.000 \text{ b/s} = 1875 \text{ KB/s}$. Luego, $\frac{40.000 \text{ ciclos}}{1s} \Rightarrow$ tiempo de transferencia = $25 \mu s \times 8 = 200 \mu s = 0,2 \text{ ms} < 0,4 \text{ ms} \Rightarrow$ el protocolo serie es más rápido

c) PAR: f = 5 KHz \Rightarrow máx cap = $\frac{\text{bits trans.}}{\text{ciclos trans.}} \times f = \frac{3b}{4 \text{ ciclos}} \times \frac{5.000 \text{ ciclos}}{1s} = 3750 \text{ b/s} = 468,75 \text{ B/s}$. Luego, $\frac{5.000 \text{ ciclos}}{1s} \Rightarrow$ tiempo de transferencia = $0,2 \text{ ms} \times 4 = 0,8 \text{ ms}$

SERIE: f = 50 KHz \Rightarrow máx cap = $\frac{\text{bits trans.}}{\text{ciclos trans.}} \times f = \frac{3b}{8 \text{ ciclos}} \times \frac{50.000 \text{ ciclos}}{1s} = 18750 \text{ b/s} \approx 2,28 \text{ KB/s}$. Luego, $\frac{50.000 \text{ ciclos}}{1s} \Rightarrow$ tiempo de transferencia = $20 \mu s \times 8 = 160 \mu s = 0,16 \text{ ms}$

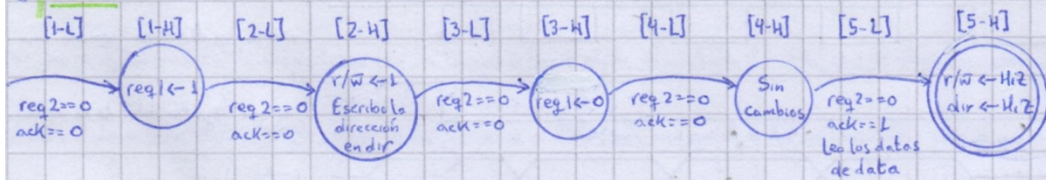
0,16 ms < 0,8 ms \Rightarrow el protocolo serie es más rápido

Ejercicio 11:



En el diagrama puede verse que como ambos procesadores desean usar el bus simultáneamente, aquel correspondiente a la señal $reg2$ debe bajarla para que el otro procesador realice la operación antes. En su vez, la señal r/w comienza y termina en alto impedancia ya que hay más de un dispositivo que puede ser master en una transferencia. Véase además que la señal $reg1$ debe estar en 1 durante toda la transferencia para que el procesador correspondiente a $reg2$ detecte el momento en que este habilitado para realizar una operación.

b) CPU1:



CPU2:

